Software-Defined Application Visibility and Control

Information About Software-Defined Application Visibility and Control

Software-Defined Application Visibility and Control (SD-AVC) is a network-level AVC controller that aggregates application data from multiple devices and sources and provides composite application information.

SD-AVC collects application data from across the network and deploys protocol pack updates in a centralized manner. SD-AVC recognizes most enterprise network traffic and provides analytics, visibility, and telemetry into the network application recognition. SD-AVC profiles all the endpoints (including wireless bridged virtual machines) connected to the access nodes to perform anomaly detection operations, such as Network Address Translation (NAT). SD-AVC can discover and alert when the same MAC address is used simultaneously on different networks.

You can enable the Software-Defined Application Visibility and Control feature on a per-WLAN basis. Also, you can turn on and turn off the Software-Defined Application Visibility and Control functionalities independently.

Note If the SD-AVC process (stilepd) crashes, Capwapd process restart or AP reload is required to resume the SD-AVC operation.

Enabling Software-Defined Application Visibility and Control on a WLAN (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless profile policy** *policy-name* **Example:** `Device(config)# wireless profile policy test-policy-profile`	Configures WLAN policy profile and enters wireless policy configuration mode.
Step 3	**no central switching** **Example:** `Device(config-wireless-policy)# no central switching`	Disables central switching and enables local switching.
Step 4	**ip nbar protocol-discovery** **Example:** `Device(config-wireless-policy)# ip nbar protocol-discovery`	Enables application recognition on the wireless policy profile by activating the NBAR2 engine.
Step 5	**end** **Example:** `Device(config-wireless-policy)# end`	Exits wireless policy configuration mode and returns to privileged EXEC mode.

Configuring Software-Defined Application Visibility and Control Global Parameters (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**avc sd-service** **Example:** `Device(config-sd-service)# avc sd-service`	Enables SD-AVC and enters software-definition service configuration mode.
Step 3	**segment** *segment-name* **Example:** `Device(config-sd-service)# segment AppRecognition`	Configures a segment name identifying a group of devices sharing the same application services.
Step 4	**controller** **Example:** `Device(config-sd-service)# controller`	Enters SD service controller configuration mode to configure connectivity parameters.
Step 5	**address** *ip-address* **Example:** `Device(config-sd-service-controller)# address 209.165.201.0`	Configures controller IP address. Supports only IPv4 address.
Step 6	**destination-ports sensor-exporter** *value* **Example:** `Device(config-sd-service-controller)# destination-ports sensor-exporter 21730`	Configures the destination port for communicating with the controller.
Step 7	**dscp** *dscp-value* **Example:** `Device(config-sd-service-controller)# dscp 16`	Enables DSCP marking.
Step 8	**source-interface** *interface interface-number* **Example:** `Device(config-sd-service-controller)# source-interface GigabitEthernet21`	Configures source interface for communicating with the controller.
Step 9	**transport application-updates https url-prefix** *url-prefix-name* **Example:** `Device(config-sd-service-controller)# transport application-updates https url-prefix cisco`	Configures transport protocols for communicating with the controller.
Step 10	**vrf** *vrf-name* **Example:** `Device(config-sd-service-controller)# vrf doc-test`	Associates the VRF with the source interface.

	Command or Action	Purpose
Step 11	**end** **Example:** `Device(config-sd-service-controller)# end`	Exits the SD service controller configuration mode and enters privileged EXEC mode.